IQS- Querying System Natural Language Processing

Likitha.M.P
Department of ISE
GSSSIETW,Mysuru
likithamp06@gmail.com

Kusumitha.N
Department of ISE
GSSSIETW,Mysuru
Kusumithakn555@gmail.com

Syeda Sukaina Ftima
Department of ISE
GSSSIETW,Mysuru
fsyeda917@gmail.com

Medha.R
Department of ISE
GSSSIETW,Mysuru
medhar099@gmail.com

Dr.Reshma Banu
Head & Prof
Department of ISE
GSSSIETW,Mysuru
hodise@gsss.edu.in

Ayesha Taranum
Asst Prof
Department of ISE
GSSSIETW,Mysuru
ayeshataranum@gsss.edu.in

ABSTRACT

The aim of this paper is to report an intelligent query system that provides and uses information from the local records. It is used to remotely retrieve related information from external sources, in order to supplement the existing records. Modern databases contain an enormous amount of information stored in a structured format. This information is processed to acquire knowledge. However, the process of information extraction from a Database system is cumbersome for non-expert users as it requires an extensive knowledge of DBMS languages. Therefore, an inevitable need arises to bridge the gap between user requirements and the provision of simple information retrieval system whereby the role of a specialized Database Administrator is annulled.

In this paper, we propose a methodology for building Intelligent Querying System (IQS) by which a user can fire queries in his own (natural) language. The system first parses the input sentences and then generates SQL queries and in turn mapped with the desired information to generate the required output. Hence, it makes the information retrieval process simple, effective and reliable.

I. INTRODUCTION

Databases are formed with the goal to facilitate the works like processing, data storage and retrieval associated with data management in information devices. The Structured Query Language (SQL) protocols are generally used in all kind of languages and relational database systems, these protocols are based on the Boolean interpretation and queries. Nowadays, there is a huge demand for the users to extract information from a database without being mandated to learn standard query languages. This will definitely provide the communication gap between the users and information storing databases.

The interaction between the systems and users in Natural Language despite of using the Querying language has led to the development of Natural Language interface to database systems. The system that enables the users to attain flexibility while querying databases is NLIDB.

Researchers at School of veterinary and Biomedical Sciences use this valuable data for research and analysis. At present when researches would like to obtain the data from the veterinary hospitals database for their research, they have to go to MUVH IT support and ask for data or records based on particular criteria or keywords.

However, graphical interfaces and form based interfaces are easier to use by occasional users, stills, invoking forms, linking frames, selecting restrictions from menus, etc. in contrast, an ideal NLIDB would allow queries to be formulated in users native language. NLIDB would be more suitable for occasional uses as there would be no need for them to spend time in learning the systems communication language.

TAGSET

On the whole lot there was a20 tags. To map the noun to attribute of table in database and verb to a relation. The table is composed of 8 tables and 4 relations between them .tables like users, teach, courses etc .and relations like teach ,registers ,marks scored etc .are present. The tags are considered as the relation between the tables also. The illustrative example shows the table tag-set of course management**.**

Table I
TAG-SET

Tag	Explanation	Example tokens
Users.name	The name of the registered User	Prof. John Smith, Dr. Rahul
Users.roll	The roll number of the registered User	200925008, 201156078
Users.stream	The study branch of the User	CSE(Computer Science)
Users.batch	The batch in which the student is studying.	ug1, ug2, pg1
Users.email	The email-id of the registered User	john@iiit.ac.in
Courses.name	The name of the Course	NLP , C Programming
Courses.code	The code of the Course	ICS231 , IEG345
Courses.credits	The number of credits of the Course	1 credit ,2 credits, 3 credits
Courses.type	The type of elcetives	Humanities, Engineering
Semester.name	The name of the semester	Monsoon, Spring
Semester.year	The year of the semester	2009, 2013, 2008
Assignment.name	The name of the assignment	Assignment 1, Assgn 2
Teach	The relation between faculty and courses	teaches, takes,taught, offers
Register	The relation between student and courses	registers, takes
Course_TA	The relation capturing Teaching Assistants	is (eg. who *is* the ta)
Course_Marks	The relation capturing the course marks	scored
Teach.ov	Overview of course in the course page	overview (eg. *overview* of NLP)
Course_Marks.marks	The marks attribute of the Course_Marks relation.	23, 34
Binary	The tag which marks yes/no question tokens	Is, do, does
Count	The tag for tokens which indicate number	How many, What number

The above ER diagram shows two entities,users adnd courses .Users represents all the users of courses portal.Users can be of two types,faculty and student.between users and courses two relations exist.
Users:Users.name,Users.roll,Users.batch and Users.email

Courses:courses.name,courses.code,courses.credits and courses.type.
Relation tags:tech and register.

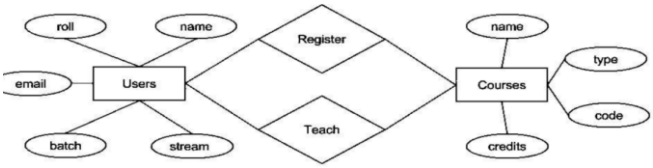

Figure 1. Sample Schema

II. OVERVIEW

Traditional querying on the database to fetch results requires knowledge of SQLs. A layman without any knowledge of RDBMS and without help from DBAs should be able to extract information from data warehouse. This write-up is the way forward and gives solution to such users who do not have any knowledge of SQL.

Understanding the Database with complex structures, varied tables, queries and relationships will be a challenge. Hence for easy and fast retrieval of data, a system in which user can input natural language queries is proposed.

RDBMS is a system that lets us create, update and administer the relational database. SQL is used to access RDBMS and data. Information about attribute names, data types, table structures and table relationships will be defined. A Semantic Builder is used to extract information from the relational database. Lexicon which is a list of all possible synonyms for attributes is made use of. Eg. Attribute name 'Educationist' comprises of 'Teacher', 'Lecturer', 'Professor' and a Lexicon will have all these synonyms. To generate a lexicon, database elements are extracted and then these elements are split into individual words. WordNet is used to recognize the possible synonyms for given set of words.

A semantic map which consists of lexicon and information about the relational database like tables and their relationships is auto generated. Semantic builder works intelligently to extract and process the information so as to intensify the semantic map. For a particular database, only one semantic map has to be created and auto Semantic Map Generator is used. Intelligent engine is used for tokenization and parsing. Eg. If the input phrase from the user is "What are the different branches of engineering ?", tokenizer produces the tokens as {What, are, the different, branches, of, engineering}. These tokens are parsed and relationships are established.

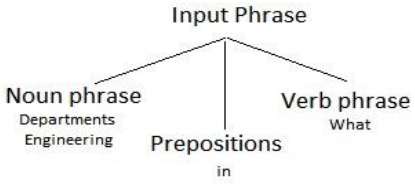

Fig2: Tokenizing and Parsing

In the second stage, the system requires the translation of queries into an intermediate code. This intermediate code is termed as Meaningful Representation. MR Generator takes tokens and generates corresponding MR map. Once MR map is created, MR checker evaluates and
Verifies its correctness.

In the last stage, SQL query is generated and presents the extracted information in a meaningful format like graph, table, etc.

III. PRESENT SCENARIO

Traditional approach uses 'Semantic Analysis' as the base between Natural Language Interface and Database Systems. The Natural Language input given by the user is parsed using semantic grammar. It creates a representation of the semantics (or meaning) of the sentence. A database query can be generated using any RDBMS and depending on the requirement output can generated.

The advantage of this approach is that we get tailor-made grammar for each database. There could be automatic generation of an NLP system for databases but, in almost all cases, the database does not have sufficient information to create a reliable NLP system. To handle all possible questions, additional information about what the data in the database represents, needs to be provided to create NLP systems.

IV. PROPOSED SYSTEM

- The purpose is to provide data processing power to non-technical personnel, who constantly rely on Database Administrators to process data. This requires the data to be processed using Natural Language Processing.
- End Users are those who deals with big chunks of data that need to be queried in order to extract the desired output. The Reading Section provides people with the data (stored in a heap) and pulls out a particular set of characteristics of data.

- The scope of the system is as follows:

  1. To work with RDBMS, one should know the syntax of the commands of that particular database software (Oracle, Microsoft SQL, etc.).
  2. Here, the Natural Language Processing is done in English i.e. the input statements have to be in English.
  3. User input is entered in an interrogative format-what, who, where, when and why.

The system proposed will include the following modules:

- GUI: Designing the Front-End or the User Interface where the user will enter the query in Natural Language.
- Parsing: The program, usually part of a compiler that receives input in the form of sequential source program instructions, interactive online commands,

markup tags or some other defined interfaces; and breaks them into tokens.

- Query Generation: Once the user statements are successfully parsed, the query is generated in SQL and is given to the back-end database
- Data Maneuvering: The output, so generated, is collected by this module. It is, then, placed in the User Interface Screen as the result.

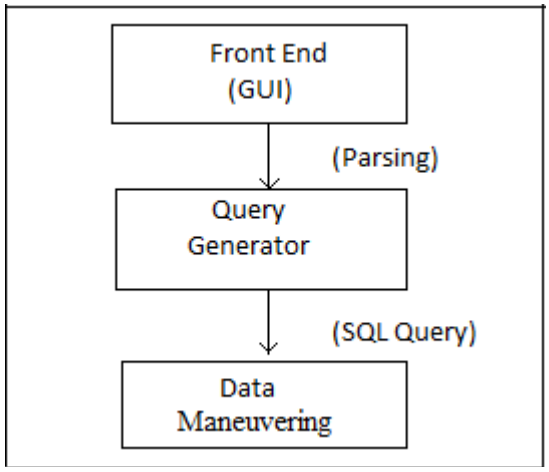

Fig3: Different Modules of proposed system

The proposed system is expected to possess a better frontend in terms of User Interface with additional modules like Speech to Text and vice-versa. Additionally, there will be a module for complex query handling based on foreign key concept.

V. SYSTEM ARCHITECTURE

Intelligent Querying System is amalgamated in an absolute manner that has several definite features. The step processes are of three types namely Semantic Building process, MR Generation process and Query Generation process.

These three process steps can be described as follows:

A. Semantic Building process

Semantic is all about the interpretation of step by step procedural study of linguistic comments. Analysis of Semantics is one in which the representations are generated and are plot to linguistic inputs.

Semantic constructors work on layers of semantic and processes and withdraw the information to build semantic map.

B. M R Generation process

The utmost goal of the system is to effectively interpret and translate Natural language processing into Structural querying language. Token relationships are mainly shown in this process stage.

Token can be phrases or words. The Natural processing language inputs are divided into tokens. Tokens are of three types: reserved keywords, attributes, alpha numeric values.

C. Query Generation process

Query Generation process involves plotting of Semantic with MR to prompt Structural querying language.

The problem of finding semantic explication of Natural language token is data base elements. Query Generator results in a set of consisting of more than one query. Later accuracy of queries is corroborated and final set of ameliorate queries is passed and return to the user.

D. Data Mining

DBIQS implies some Data Mining and Data Warehousing techniques which allow users to create tailored views. Using these views the information can be viewed and extracted in desired ways.

Information from two or more sources can be pooled together, eliminating the need of filtering, pivoting and formatting. Abstract views can be created and information can be represented directly in the form of spreadsheets including *tables, bar graphs and pie charts.*

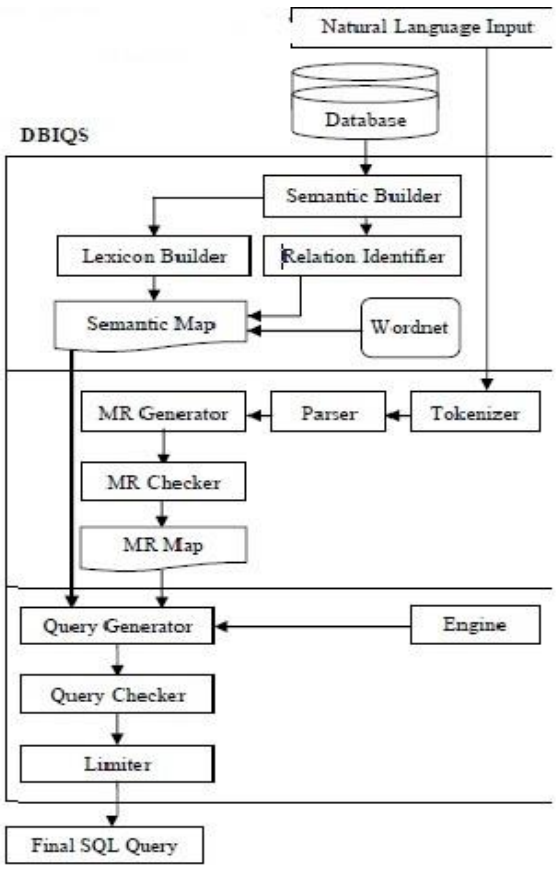

Fig 4:DBIQS Architecture

VI. ANALYSIS

DBIQS proves to be highly efficient in producing SQL queries from the tractable natural language queries and also portrays a very high proficiency in translating intractable queries to partially tractable or completely tractable queries.

From a set of 150 tractable queries developed over an student database containing information related to student's performance and departments etc., DBIQS produced highly accurate results.

VII. CONCLUSION

This system presents a pragmatic solution for non-expert users to query a database in the form of natural language questions. IQS exhibits a high level of efficiency in generating precise and accurate queries. This is possible because it proficiently translates intractable queries to partially tractable or completely tractable forms.

It also enables the users to access the information in most feasible form by creating tailored views of the results. It can represent information in the form of tablets, graphs and pie charts.

In future, IQS can be integrated with advanced data interactions tools, multi-lingual NLIDBs and deep learning techniques to extract higher throughput from the system.

REFERENCES

[1] I. Androutsopoulos, G. D. Ritchie and P. Thanisch, "Natural Language Interfaces to Databases – An Introduction," in Natural Language Engineering, vol. 1, part 1, 1995, pp. 29-81.

[2] George A. Miller (1995), "WordNet: A Lexical Database for English," in Communications of the ACM Vol. 38, No. 11: 39-41.
[3] Christiane Fellbaum (1998, ed.), "WordNet: An Electronic Lexical Database," Cambridge, MA: MIT Press.

[4] Ana-Maria Popescu, Oren Etzioni and Henry Kautz, "Toward a Theory of Natural Language Interfaces to Databases," in IUI Proceedings 2003.

[5] Ana-Maria Popescu, Alex Armanasu, Alexander Yates, Oren Etzioni and David Ko, "Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability," in COLING 2004.

[6] Yuk Wah Wong and Raymond J. Mooney, "Learning for Semantic Parsing with Statistical Machine Translation," in Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL-2006), NY, 2006, pp. 439-446.

[7] Yuk Wah Wong and Raymond J. Mooney, "Learning for Semantic Parsing with Ststistical Machine Translation Techniques," Technical Report (UT-AI-05-323), Artificial Intelligence Lab, University of Texas at Austin, Austin, TX, 2005.

[8] Norman Fairclough, "Discourse and Text: Linguistic and Intertextual Analysis within Discourse Analysis," Lancaster University, Discourse Society, April 1992 vol. 3 no. 2, pp. 193-217.

[9] J. Allen, "Recognizing Intentions from Natural Language Utterances," in Michael Brady and Robert C. Berwick, editors, "Computational Models of Discourse", chapter 2, MIT Press Cambridge, Massachuttes, 1983 pp. 107-166.